2016 International Conference on Distributed Computing in Sensor Systems

# Round Trip Time based Adaptive Congestion Control with CoAP for Sensor Network

Jung June Lee, Sung Min Chung, Byungjun Lee, Kyung Tae Kim, Hee Yong Youn
College of Software
Sungkyunkwan University
Suwon, Korea
{jungjune86, gearfried, byungjun}@skku.edu, kyungtaekim76@gmail.com, youn7147@skku.edu

***Abstract*—Constrained Application Protocol (CoAP) was developed to support the communication between resource constrained nodes via low-power links. As an Internet protocol, CoAP needs congestion control primarily to stabilize the networking operation. In this paper we propose a new round trip time based adaptive congestion control scheme, which improves CoAP by utilizing the retransmission count information in estimating the retransmission timeout. An experiment is conducted based on Californium CoAP framework and real devices. It shows that the proposed scheme significantly improves CoAP in terms of throughput and rate of successful transaction.**

***Keywords-Adaptive Congestion Control; CoAP; Round Trip Time; Retransmission Timeout; Sensor Network***

## I. INTRODUCTION

Wireless sensor network is now widely used in which a large number of sensor nodes are interconnected to collect and process the data from the target area. Typically, the nodes operate in the presence of lossy and noisy communication links [1]. CoAP is defined in a standard RFC [2] by a working group of IETF called Constrained RESTful Environment. CoAP uses the same RESTful principle as HTTP, but it is much lighter to be run on constrained devices. The CoAP interaction is based on client/server model. A client sends a request specified by a method code (GET, PUT, POST or DELETE) on the resource (identified by a URI) of the server. The CoAP server returns a message containing response code and payload. CoAP handles the interactions asynchronously over a datagram-oriented transport like UDP.

One of the most important constraints on IoT is resource limitation of the nodes, causing network congestion [3]. CoAP RFC defines a basic congestion control mechanism, which doubles the retransmission timeout (RTO) value when timeout occurs. Here one issue is that the network manager has to set up the RTO value manually. As an enhancement, the IETF CoRE working group prepared a draft specification called CoAP Congestion Control Advanced (CoCoA) [4], in which the round trip time (RTT) is used to set the RTO value automatically. The earlier research shows that the performance of CoCoA is better than CoAP in congested networks [5]. However, it still has a limitation in setting a right RTO value with burst traffic. This is because correct RTT of retransmitted packet is hardly obtained which is an important factor in deciding the RTO value.

The main hurdle in obtaining the RTT of retransmitted packet is that the origin of a response packet is not available. To resolve this problem, we propose an advanced congestion control mechanism for CoAP which utilizes the retransmission count of the request packet. The count information is used to match the request and response packet and calculate actual RTT. The simulation results show that the proposed scheme can effectively increase the throughput of the network and reduce retry rate in comparison with CoAP. With 100 clients, the proposed scheme allows about 150% higher throughput than CoAP.

## II. RELATED WORK

CoAP provides a request/response interaction model between the endpoints of the application. Confirmable (CON) message is used for request, and the response is carried in the Acknowledgement (ACK) message which includes the ID of request message. CoAP RFC defines this flow as "Piggybacked Response" [2].

In the low-power and lossy network environment, it is common that unexpected packet loss or packet delay occurs [6]. To handle these problems, CoAP provides basic congestion control based on an exponential back-off mechanism. After transmitting the CON message, if the sender does not receive an ACK in the interval of RTO value, CON message is retransmitted. The format and data of retransmitted message are same as the initial one, while the RTO value is doubled and retransmission count is incremented. The sender keeps retransmitting the CON message until the retransmission count reaches a limit, with 4 as the default value.

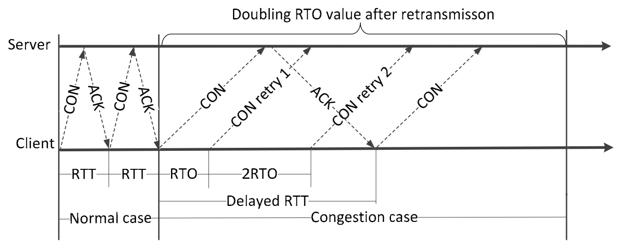


Figure 1. The default congestion control of CoAP

Figure 1 shows the default congestion control mechanism of CoAP. Normally, the client sends CON right after receiving the ACK message, implying that the interval of CON is same as RTT. If congestion occurs, the interval increases until the client receives the ACK message. Because the default scheme doubles RTO each time, the client may need to retransmit CON several times until right RTO value is found. In the worst case, the client retransmits packet whenever RTO is less than RTT.

## III. THE PROPOSED SCHEME

To handle the RTO update problem, we propose to employ retransmission count to identify correct RTT of every packet via packet matching. Here RTOstrong estimator is used for each request packet, which is accurate even with burst traffic. Fig 2 shows the operation flow of the proposed scheme.

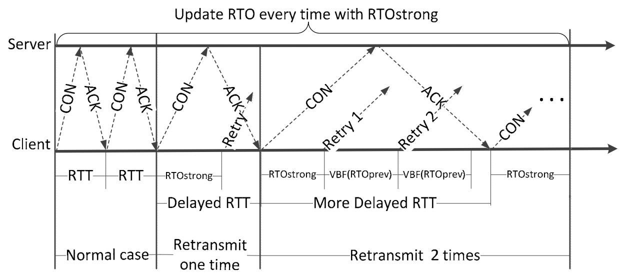


Figure 2. The flow of congestion control with the propsoed scheme.

The main difficulty of CoAP in obtaining actual RTT of retransmitted packet is that the origin of response packet cannot be identified. When a request is retransmitted, it has same message ID and data as the initial one. Also, the response packet generated by the piggyback method has the same message ID as the request. Figure 3 compares the transaction occurring in congested network with CoAP and the proposed scheme. Observer from Fig. 3(a) that Device-A cannot obtain correct RTT because there is no information in the final ACK packet corresponding to the second CON request.

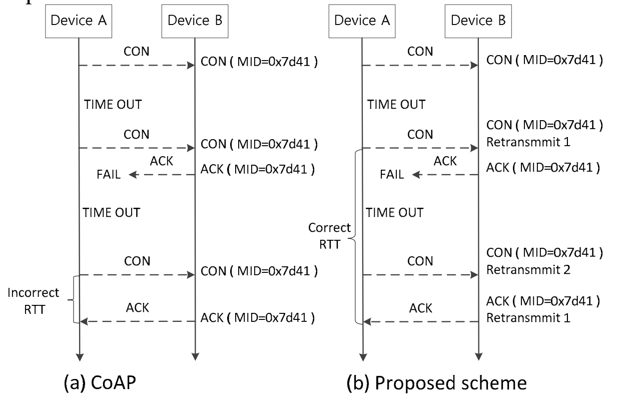


Figure 3. The comparision of retransmission flows in congested network

To identify the origin of a request packet, the retransmission count is used in the proposed scheme as shown in Fig. 3(b). When retransmission occurs at Device-A, retransmission count is put in the request packet. Device-B generates ACK packet with the copied retransmission count based on the piggyback mechanism. After receiving ACK packet, Device-A can identify the origin of ACK and obtain correct RTT. To minimize the operation overhead, the count is put in only the retransmitted request.

## IV. PERFORMANCE EVALUATION

To properly evaluate the proposed congestion control scheme, the devices of limited resource and congested network are considered. For this, Raspberry Pi is used as a server to generate the congestion condition. The specification of the server is 700 MHz CPU, 512 MB SDRAM, 10/100 BaseT Ethernet. For the client, multiple clients are implemented based on threads. The specification of client PC is 3.2 GHz quad core CPU, 8 GB RAM. Between the client and server, UDP is used for Ethernet communication.

The experiment is conducted with the open source library of CoAP. The performance metrics adopted in the experiment include the throughput and rate of successful communication. If a proper RTO value is set, the communication will be successful without any or few retransmissions. First, congestion due to the requests simultaneously generated by multiple clients is considered. For this, 10 request packets are generated for each client thread, while the number of threads is increased from 1 to 100. Here every client employs the same congestion control mechanism.

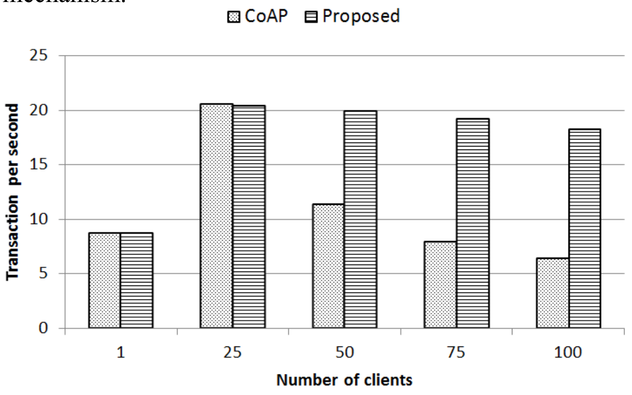


Figure 4. The comparison of throughputs

Figure 4 compares the throughput of the schemes. When the number of clients is smaller than 25, congestion does not occur, and almost every communication is successful. Some packets might be retransmitted with the proposed scheme even when the network is stable due to dynamic RTO estimation. RTT usually fluctuates and thus RTO needs to be updated which may cause retransmission and decrease the throughput. If RTO is fixed like CoAP, there is no retransmission until RTT becomes less than RTO. Note that CoAP was designed for constrained environment of limited resource and unstable network. Therefore, the performance of the schemes needs to be tested for congested network, which occur with more than 50 clients. With congested network, the proposed scheme displays much better performance than CoAP. With 100 clients, it allows about

150% higher throughput than CoAP. Also notice that the throughput of CoAP drops dramatically as the number of clients increases while the decrease is small with the proposed scheme.

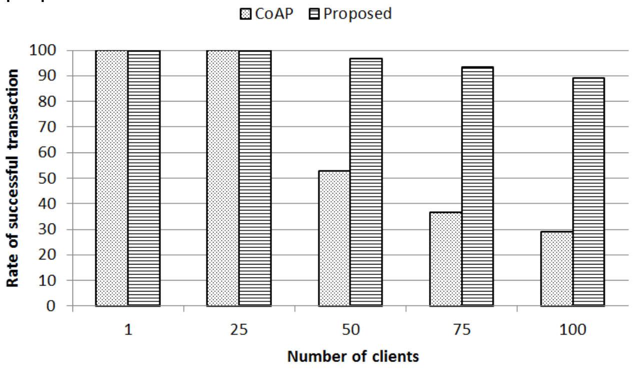


Figure 5. The flow of retransmission in congestred network

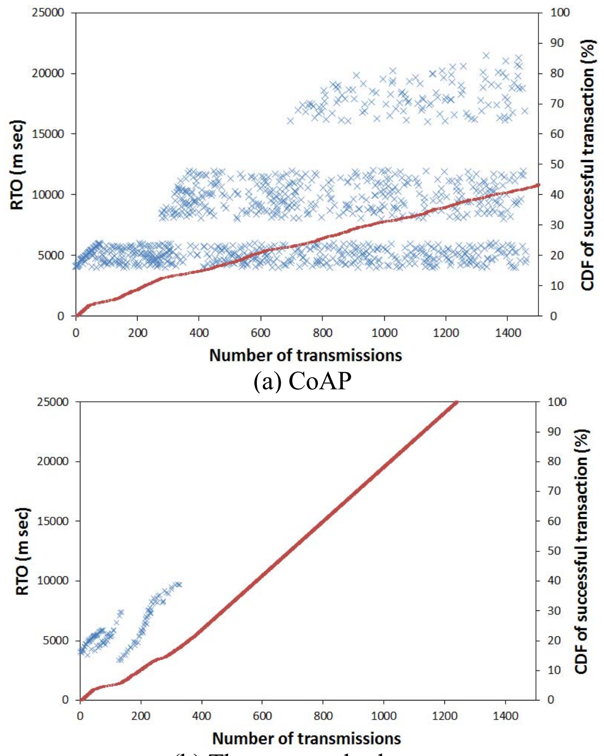
Figure 5 compares the rate of successful transaction. Higher rate of successful transaction implies less retransmission during the communication. Rapid drop in the rate of CoAP with increasing number of clients reveals that fixed RTO value is not effective in handling congested network. Notice that the proposed approach is quite effective in improving CoAP.

The distributions of retransmissions and RTO value of the two schemes with 100 clients are compared in Figure 6. Here, the cumulative distribution function (CDF) of successful transactions is shown for the given number of transactions [5]. During the experiment with 100 clients, some communications may not be successful, and thus 1500 transaction logs are checked for each scheme. The RTO value is limited to 25000 milliseconds. In the graphs the solid line and 'x' mark represent the CDF and RTO, respectively. Observe from Figure 6(a) that three groups of RTOs exist as the bottom group of first RTOs and the top one the third RTOs. Notice that fixed RTO causes a large number of retransmissions. On the other hand, Figure 6(b) reveals the effectiveness of the proposed scheme based on RTO adaptation. A proper RTO value is found under 400 transactions and the rate of CDF becomes almost linear afterwards.

## V. Conclusion

In this paper we have presented a novel congestion control scheme for CoAP based on the RTT. Here correct RTT is obtained using the option field of the message for holding retransmission count, and thus the proposed scheme can be implemented without any conflict with the existing protocol and extra overhead. An experiment with actual IoT device reveals that the proposed scheme significantly improves the performance of CoAP in terms of throughput and the rate of successful transaction.

Congestion control is a crucial issue for CoAP since the deployed nodes have limited resource. Continuous retransmissions make the congestion condition worse, and reduce the life time of IoT devices. In the future more comprehensive performance model based on queuing theory will be developed for the proposed scheme. Enhancement of the proposed scheme in the RTO estimation will also be conducted, along with the comparison with CoCoA.



(b) The proposed scheme

Figure 6. The RTO values and CDF of succesful transaction with 100 clients

## Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2014R1A1A2040257 and 2014R1A1A2060398), the second Brain Korea 21 PLUS project and Samsung Electronics.
Corresponding author: Hee Yong Youn

## References

[1] Ala Al-Fuqaha et al., "Internet of Things A Survey on Enabling Technologies, Protocols and Applications," *Communications Surveys & Tutorials*, pp. 1, June 2015.

[2] Z. Shelby et al., "The Constrained Application Protocol (CoAP)," *IETF RFC 7252*. June 2014.

[3] Hu Yuan et al., "Congestion Control for Wireless Sensor Networks: A survey," *Control and Decision Conference*, pp 4853-4858, May 2014.

[4] C. Bormann et al., "CoAP simple Congestion Control/Advanced," *IETF I-D draft-bormann-core-cocoa-02*, January 2015.

[5] A. Betzler et al., "Congestion Control for CoAP Cloud Services," *Emerging Technology and Factory Automation*, pp 1-6, September 2014.

[6] J. Kaur et al., "A survey on recent congestion control schemes in wireless sensor network," *Advance Computing Conference*, pp 387-392, June 2015